

WHAT IS CLAIMED IS:

1. A switch for transferring data comprising:

at least one master unit;

a plurality of slave units;

a bus through which the master unit communicates with the slave units; and

a memory in communication with the master unit having a software program which causes the switch to automatically recover when a slave unit fails.

2. A switch as described in Claim 1 including persistent storage that survives across abnormal termination of the switch.

3. A switch as described in Claim 2 including a mechanism for detecting failures of the slave units and thereupon causes the switch to abnormally terminate.

4. A switch as described in Claim 3 wherein the software program causes the switch to automatically recover when the detecting mechanism causes the switch to abnormally terminate.

5. A switch as described in Claim 4 wherein the detecting mechanism includes a hardware watchdog device.

6. A method for transferring data comprising the steps of:

attempting to access a failed slave unit of a plurality of slave units of a switch by a master unit of the switch with a signal through a bus through which the master unit and the failed slave unit communicate; and

automatically recovering the switch from the failed slave unit with a software program in the switch that directs the master unit to avoid further accessing the failed slave unit of the plurality of slave units.

7. A method as described in Claim 6 wherein the recovering step includes the step of obtaining status information about the slave units from persistent storage.

8. A software program comprising the steps of:

identifying a first slave unit of a plurality of slave units of a switch has failed when the first slave unit is attempted to be accessed by a master unit of the switch; and

preventing a master unit from attempting to access the failed first slave unit.

9. A software program as described in Claim 8 including the step of determining the switch abnormally terminated when the master unit attempted to access the first slave unit.

10. A program as described in Claim 9 including the step of changing information in persistent storage associated with the first slave unit from identified as failed to identified as good if the switch does not terminate abnormally after the master unit attempts to contact the slave unit.

11. A program as described in Claim 10 including the step of setting a variable slot chosen from amongst a plurality of slots of the switch not marked as potentially bad.

12. A program as described in Claim 11 including the step of determining whether the first slave unit is physically present in a first slot of the plurality of slots.

13. A program as described in Claim 12 including the step of determining the first slot is marked to be skipped.

14. A program as described in Claim 13 including the step of marking the variable slot as potentially bad if it is not marked potentially bad.

15. A program as described in Claim 14 including the step of reporting the variable slot as containing broken hardware and preventing the master unit from attempting to access the variable slot if the variable slot is marked to be skipped.

16. A program as described in Claim 15 including the step of attempting to access hardware present in the variable slot if the variable slot is marked potentially bad.

17. A program as described in Claim 16 including the step of marking the variable slot as good if the switch did not abnormally terminate when the master unit accessed the first slave unit.

18. A program as described in Claim 17 including the step of enabling normal operations on hardware present in the variable slot if the variable slot is marked as good.

19. A program as described in Claim 18 including the step of setting the variable slot to a next slot of the plurality of slots.